Location Detection Using Wifi

By

Mohd Khair Bin Ali (4451)

Desertion submitted in partial fulfillment of
the requirements for the
Bachelor of Technology (Hons)
(Business Information System)

OCTOBER 2006

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Location Detection Using Wifi

By

Mohd Khair Bin Ali

A project dissertation submitted to the
Business Information System Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
BACHELOR OF TECHNOLOGY (Hons)
(BUSINESS INFORMATION SYSTEM)

Approved by,

Mr. Abdullah Sani Abdul Rahman

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
October 2006

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

MOHD KHAIR BIN ALI

ABSTRACT

There are many approaches for indoor location detection such as using infrared and ultra sound. Beside that, another powerful approach for indoor location detection is RF (Radio Frequency). In emergency situation happened in a building such as fires, it is difficult to make sure all people in the building have left out. It is even worst if there are victim trapped but no body know where the exactly location of the victims. The project is construct a network environment based on wireless LAN infrastructure and supports the protocol over the 802.11g. The system is organized by client-server, access points and PDAs or laptops as client devices. Location determination method is implemented on the basis of signal strength. The test program developed using the sample provided able to detect the access points available. The parameter that can be get using the program including the SSID, MAC address, Network type, WEP, RSSI and etc. To identify the distance testing has been made based on the RSSI gathered. The testing is made by calculating manually the distance of the location based on the RSSI collected. The proposed system is an indoor positioning system based on signal strength measurements, which were approximated by the received RSSI in a mobile device. The triangulation method combined with distance estimation is used to predict the position of the terminal. It is hoped that with some improvement to the proposed system able to work in the emergency condition.

ACKNOWLEDGEMENTS

First of all I would like to express my gratefulness to Allah for lending me the strength to accomplish my FYP project.

I would also like to extend my sincere gratitude to my UTP supervisors, Mr. Abdullah Sani for the courtesy of supervising me during my final year project period. Next, thank you to my parents and family members for the support given to me. Finally thanks to Hassan Bin Sabirin, Mohd Juzaili, and to all people that were directly or indirectly involved in my effort to complete my Final Year Project.

Thank You.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND NOMENCLATURES

AP	Access Point
CCK	Complimentary Code Keying
DDK	Driver Development Kit
IR	Infra Red
IEEE	Institute of Electrical and Electronics Engineers
LAN	Local Area Network
OFDM	Orthogonal Frequency-division Multiplexing
RSSI	Receive Signal Strength Indicator
RF	Radio Frequency

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND OF STUDY

There are many approaches for indoor location detection such as using infrared and ultra sound. Another powerful approach for indoor location detection is RF (Radio Frequency). This approach is more suitable for this project because of its inherent ability to penetrate many types of surfaces, especially at lower frequencies. In addition, RF-based indoor location detection has significant cost and maintenance benefits, as it can be readily deployed over existing radio infrastructures, such as IEEE 802.11 wireless LANs.

1.2 PROBLEM STATEMENT

In emergency situation happened in a building such as fires, it is difficult to make sure all people in the building have left out. It is even worst if there are victim trapped but no body know where the exactly location of the victims. Moreover the potential hazarded happened, or the source of emergency occurred in the building also cannot be detected to allow the immediate response. Thus it surely will delay all the rescue operation to be completed.

1.3 OBJECTIVE

The objectives of this project are threefold:

- To enable crew members to identify their own locations.
- To locate victims, potential hazards, or sources of the emergency.
- To identify trapped personnel.

CHAPTER 2

LITERATURE REVIEW AND/OR THEORY

From the author's research, it is discovered that there are a number of previous works that has been done that relates to the location detection system.

Current Trends in Wireless Technology - Wireless LAN Hotspots

Among the growing number of modern trends in the arena of wireless technology is the use of wireless LAN hotspots. Webopedia defines a wireless hotspot as "*A specific geographic location in which an access point provides public wireless broadband network services to mobile visitors through a WLAN. Hotspots are often located in heavily populated places such as airports, train stations, libraries, marinas, conventions centre and hotels. Hotspots typically have a short range of access.*" (www.webopedia.com 2003)

From this definition we can see that a hotspot is basically a physical area of wireless network services based around one or more access points. To understand the basis of the term "hotspot", it is first necessary to look at the functionality of an access point. An access point can be defined as "*a hardware device or a computer's software that acts as a communication hub for users of a wireless device to connect to a wired LAN. Access Points are important for providing heightened wireless security and for extending the physical range of service a wireless user has access to.*"(www.webopedia.com 2003)

Hotspots utilise the now globally accepted IEEE wireless networking standard, 802.11, to provide access to e-mail, the internet, and other data/network applications. In recent times, the capability of the 802.11 standard has led to its use in the provision of wireless network access in public areas such as pubs, hotels and coffee shops. This growing use of the 802.11 wireless standard or WiFi (Wireless Fidelity) has led to a massive consumer interest in such public amenities and has brought their customer service abilities to a new level. Customers with WiFi compliant mobile devices can now go to a hotspot encompassing a café or pub and check their e-mail or use the Internet, often at no extra charge. The benefit to the provider or owner of these facilities is the surge in consumer interest that leads to sales increases in the other previously available products.

Existing location system developed

The first system assessed was based on Infra-Red (IR) technology. According to the developers of one of these systems, IR has a definite advantage because "technology has already been exploited commercially, it is inexpensive and readily available for developing new applications." (Falcao *et al* 1992) These developers created the Active Badge system which uses wearable, specially designed IR tags to transmit location data to a network of sensors. The second IR-based system that was evaluated was the ParcTab Mobile Computing System. (Adams *et al* 1993)

This system relies on a custom built PDA that must be carried by the user. They also developed a number of ParcTab applications in order to explore the usefulness of such a system. While the system have proved successful in accurately locating users, from the research, it is identified a number of problems associated with their approach.

Initially, this approach requires a purpose built device thereby adding to the costs associated with adopting it. It also requires that the building in which it is deployed be fitted with a network of sensors. These sensors, while low in financial cost, result in even more hardware being needed.

The next systems that were evaluated were created using radio frequency (RF). The first of these, The Cricket Location-Support System, took a decentralized approach to system administration whereby a listener on the mobile device listens for announcements from a "location beacon" and uses an inference algorithm to determine its location (Chakraboty et al 2000).

Having spent some time 26 experimenting with radio signal strengths to infer distance developers of Cricket decided that it would not give an accurate enough estimation of location. With this in mind they decided to use a combination of radio and ultrasonic equipment to provide more accurate results. The resulting system can estimate the location of a mobile device with an estimated error distance of 1.5 meters (Chakraboty et al 2000).

Developers of the SpotON Ad-Hoc Location Sensing System developed tags which use received radio signal strength information (RSSI) to estimate inter-tag distance. They used algorithmic techniques to factor out erroneous measurements due to signal attenuation (Borriello et al 2001). This ad-hoc location sensing system also takes a decentralized approach to administration.

As with the IR-based systems, both Cricket and SpotON require additional hardware to be developed. The design of the Cricket system involves listeners being attached to mobile devices, software present on the mobile device then determines its location.

The next system assessed was called the RADAR system. RADAR is a location sensing system developed by a Microsoft research group. The system is deployed building-wide and is based on the IEEE 802.11 Wave LAN wireless networking technology. At the access point, or base station the signal strength is measured between it and a wireless device. This data is then computed to determine a 2D position within a building. RADAR requires a small number of base stations and uses existing wireless networking infrastructure. Disadvantages of RADAR include the requirement of mobile devices to support a wireless LAN. The 2D limitation is also a problem if one considers use of such a system in a multi-floored building.

A major advantage of this system over those others that have been evaluated is that by utilizing wireless LANs, RADAR not only provides a facility for locating mobile hosts it also provides data networking services (Bahl and Padmanabhan 2000).

Each system described above has associated limitations and flaws which are summarized in the following list:

1) Many require specific hardware to be developed, namely those systems that use IR and RF for which wearable tags or badges were needed.
2) Along with the requirement to develop these tags/badges there is also a onus on the user to ensure that the tag is about their person and is visible at all times
3) Most of these systems don't scale well because as the amount of devices to be tracked/located increases so to do noise levels, which can affect the reliability of signal strength readings and therefore affect the accuracy of the system
4) None of systems have been designed for the specific working conditions of emergency networks, and are thus generally unsuitable for this purpose.

It is identified these limitations as being major drawbacks to successful acceptance of location aware systems and in order for a system to be successful these issues must be addressed. It is therefore proposed that there exists a need for the development of a location sensing system that can:

1) Take advantage of existing infrastructure
2) Be as transparent as possible to the user
3) Be scalable so that it can be used in small, medium or large environments both indoor and outdoor

CHAPTER 3

METHODOLOGY / PROJECT WORK

3.1 PROCEDURE

This project will be executed in two phases; first semester and second semester.

First Semester

In the first semester, the project will be concentrating on research activities and revise the related works (literature review) that have been done before. It is expected that by the end of the semester, the author would be able to identify the related factors, at least on the surface, and from that the author can start narrowing down and study more on the particular factors. It is also expected by the end of the semester, the author would be able to see clearly the next task to be done, and lay out a plan that will achieve the projects objectives.

System Architecture

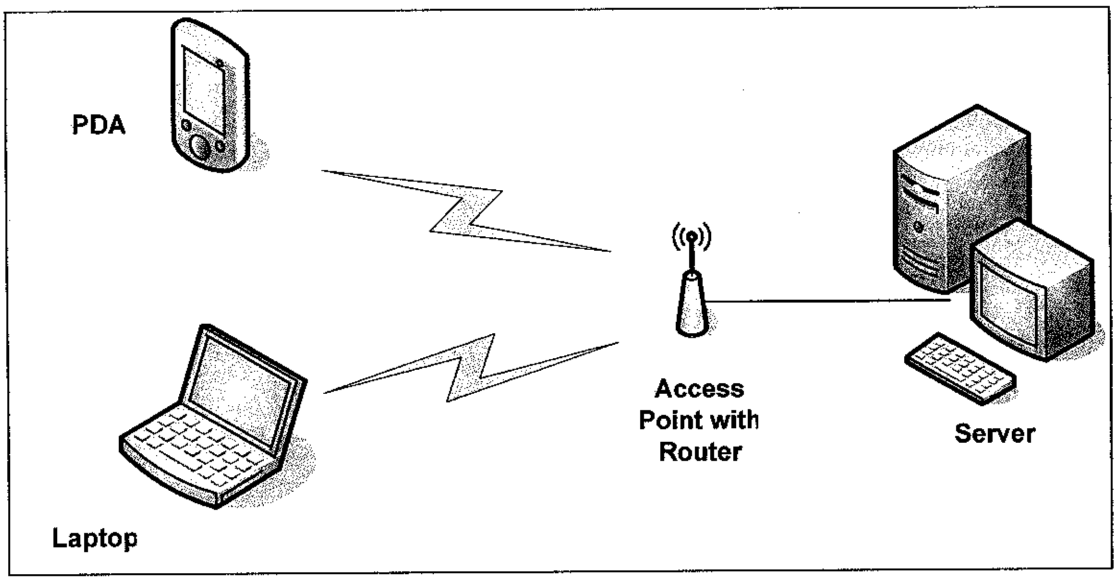

Figure 1.1: System Architecture

Network Infrastructure

Construct a network environment based on wireless LAN infrastructure and support the protocol over the 802.11g. The system is organized by client-server, access points and PDAs or laptops as client devices. Location determination method is implemented on the basis of signal strength

PDA or laptop and AP each other connect with 802.11b(Wi-Fi) and the network is composed through connecting AP to location determination server in Ethernet form. A PDA or laptop receives radio frequency signals from a neighbor AP, hold BSS IDs, and connect a communication session with AP. Use RSSI value among attributes supporting in the standard of IEEE 802.11b wireless LAN for determining location.

How Wifi Works

Computers, laptops, cell phones and palm pilots are examples of mechanisms that can grant the user internet access. Although computers and laptops are capable of having normal internet connection (i.e. Ethernet connection), they are also able to have internet access through wireless technology. Cell phones and palm pilots can only be connected to the internet by wireless connection. Wireless networking is possible through the technology of wireless-fidelity. Wireless-fidelity or WiFi as most people call it allows a ubiquitous internet connection to be broadcasted through radio waves. Its purpose serves directly to the users looking for internet access devoid of any cords or wires.

Radio waves are the keys which make WiFi networking possible. These radio signals are transmitted from antennas and routers and are picked up by WiFi receivers such as computers and cell phones that are equipped with WiFi cards. Whenever a computer receives any of the signals within the range of a WiFi network which is usually 300 – 500 feet for antennas and 100 – 150 feet for routers, the WiFi card will read the signals and thus create an internet connection between the user and the network without the use of a cord. Usually the connection speed is increases as the computer gets closer to the main source of the signal and decreases when the computer gets further away. With that in mind, think of the WiFi card as being an invisible cord that connects your computer to the antenna for a direct connection to the internet. Many new laptops already come with a WiFi card built in and in many cases we don't have to do anything to start WiFi which is one of the best things about it and how simple it is. WiFi cards can be external or internal, meaning that if a WiFi card is not installed in our computer, we may purchase a USB antenna attachment and have it externally connect to our USB port, or have an antenna-equipped expansion card installed directly to the computer. Laptops without a built in WiFi card is usually installed the external way while PCs have it installed internally. Once a connection is established between the user and the network, the user will be prompted with a login screen and password if it is a fee-based type network. Though there're also free-based network connections as well in some areas. Wifi networking around the world is creating hot spots in cities where anyone with a laptop can wirelessly plug into the internet. A hotspot is a connection point for a WiFi network. It is a small box that is hardwired into the internet. There are many WiFi hotspots now available in public places like restaurants, hotels, libraries and airports. You can also create our own hotspot in our home. Research has shown that within the next 10 years, hot spots will be all over urban and suburban vicinities.

Radio Signals

When it comes to WiFi technology, radio signals are the most essential and crucial components. They determine if you have internet access and what speed the connection is running on. The basic concept of radio signals used in WiFi networking is usually compared to that of walkie talkies. These walkie talkies can transmit and receive radio signals which makes possible to communicate with one another. When you literally talk into the walkie talkie, it is fixed to a radio frequency and broadcasted with the antenna. The person holding the other walkie talkie receives the radio transmission through its antenna and decodes the words from the original walkie talkie. Thus the same idea is used when it comes to radio signals in WiFi networking. The main sources of radio waves come from the antenna or router which transmits these waves to another antenna, in this case, WiFi cards, which give a direct signal to the user the ability to have internet access. For a better understanding, look back again at the photographic representation in the summary section. The antenna is placed on top of a building and emits radio waves within the vicinity. Notice that people with internet devices are receiving these signals to gain access to the internet. The IEEE (Institute of Electrical and Electronics Engineers) has produced a set of standards and specifications for wireless networks under the title "IEEE 802.11" that defines the format and structure of radio signals sent out by WiFi networking routers and antennas. Currently there are three forms of the 802.11 standard proposed by the IEEE: 802.11b, which came before 802.11a, and then 802.11g as its last form. WiFi radios that work with 802.11b and 802.11g broadcasts at a frequency rate of 2.4 GHz while those that comply with the 802.11a form broadcasts at 5GHz. The higher frequency allows faster data rates. Each standard has advantages, but 802.11g has the speed, compatibility, and range to replace 802.11b as the most common configuration of WiFi. Here is a chart that represents the main differences of each standard.

Table 1: Comparison of wifi standard

Standard	802.11b	802.11a	802.11g
Speed	11 Mbps	54 Mbps	54Mbps
Range	100-150 feet indoors	27-75 feet indoors	100-150 feet indoors
Frequency	2.4GHz, a band already crowded with cordless phones	5GHz, an uncrowded band	2.4GHz, still a crowd of cordless phones and microwaves
Acceptance	Hot spots are already established using 802.11b. Equipment is readily available	More common in corporate and office environments.	802.11g is compatible with the specs for 802.11b, meaning it can be used on a network based on b or g versions.

The standards 802.11a and 802.11g use much more explicit encoding techniques that contribute to the much higher data rates. This is known as orthogonal frequency-division multiplexing (OFDM). As for 802.11b, it is called Complimentary Code Keying (CCK). The three radio standards in wireless networking have the ability to change frequencies. The 802.11b cards can indirectly transmit radio signals onto any of the three bands or they can split the available bandwidth into many channels and does the term "frequency hop" between them. What's helpful about frequency hopping is that it overcomes interference from other people who are using the same radio signal and changes frequency various times per second. This allows numerous WiFi cards under the same radio signal to talk concurrently with no interference with each other.

WiFi Cards

In order for computers to receive these radio signals, a network adapter must be installed on the computer. The network adapter in this case is called the WiFi Card and it can take several physical forms. For laptops, this card will be a PCMCIA card in which you insert to the PCMCIA slot on the laptop. The other way is to buy a external adapter and plug it into a USB port. For personal computers, you can install plug-in PCI cards or a small external adapter for the USB port just like the one used for laptops. A network adaptor should be capable to use in any operating system such as WINDOWS, MAC OS, LINUX and UNIX as long as the driver for the adapter is accessible to download or install. As you already know, there are three forms of standards used for wireless networking: 802.11b, 802.11a, and 802.11g. Therefore, there are three different kinds of WiFi cards that are available to purchase. The recommended WiFi card to buy is the 802.11g because it has the advantage of higher speeds than the 802.11b (see the chart in the radio signals section). Although it is bit more expensive than the 802.11b, it is still worth the cost. A hotspot which contains 802.11 standards can hold up to as much as 100 802.11 cards within the vicinity.

Hotspots

When laptops are equipped with WiFi cards, it can very well connect to the wireless hotspot. As mentioned earlier, hot spots are connection points for WiFi networks. They are simply locations where wireless internet is available for those who have internet ready instruments (i.e. computers, laptops, cell phones and palm pilots). Once the internet connection has been established, a log-on screen will appear for most computers and laptops. This occurs when you have attempted to use a wireless network connection that is a fee-based type, meaning you must use your credit card and pay to be a member of that certain internet connection. There are also free-based type wireless connections available as well. As reliable as it sounds having internet connection without the use of cords, wireless internet is very exposed to hackers and very insecure in most cases. Hackers can track down what you are looking at and can implant viruses such as worms and Trojans and can even access your data! That is why people use firewall on their connection to prevent any harm caused by hackers. Firewall is a security device that controls access from the internet to a local network.

Access Points

Access points are often combined with other network functions. It is very likely that you will discover a separate access point that just plugs into a wired Local Area Network (LAN). If we already have more than one computer hooked together on the same network and want to have a good hotspot, you can buy a wireless access point and plug it in to the network. As mentioned earlier, radio signals can be received and transmitted by an antenna and a router. The router is an example of an access point of how multiple computers can be connected together in the same network both including the use of wires and wireless technology. So once you turn on your access point on, you will have a hotspot in your home and will have radio signal within a 100 feet radius. As for antennas, they have a higher signal transmission at a 300-500 feet radius. That is why hotels, campuses, libraries, etc, anything that is larger than your home is required to use antennas rather than routers. Regardless of its size and shape, all access points consist of a radio that transmits and receive signals and data between network stations and an Ethernet port that connects to a wired network (normal connection).

Function of the Working Parts

The radio signal, data structure, and the network structure are the three essential elements that form the wireless internet standard 802.11 WiFi networking. Radio signals of 802.11b, 802.11a, and 802.11g can be received or transmitted from antennas and routers, which are known as access points, through laptops, computers, cell phones and palm pilots. The standard 802.11g radio signal is considered to be fastest and most reliable when it comes to quality compared to the other two standards. In order for computers, laptops and palm pilots to receive this wireless connection, they must be properly equipped. Meaning a WiFi card must be installed either externally or internally to the computer, laptop or palm pilots. WiFi cards come in three different types just like how there are three different WiFi standards, the 802.11 series. Once a successful connection has been made from the access point to the WiFi card, the user may now access the internet without any cords used. The name of the term regarding the location where one can access the internet wirelessly is called a "hotspot". There are fee-typed and free-typed based wireless connections in Wifi. If a user is given a log on screen, when first connecting to wireless internet, it is most likely going to be a fee-typed base connection. Although there are areas where there are free-typed base as well. Wireless connection may sound reliable at times; it is very prone to hackers. They have could enter our personal computer data and implant viruses and worms into our computer. That is why people are being urge use firewall as a protection from these notorious hackers. Speaking of security, a WiFi connection can either be open or secure. If a secure connection exists, a user must input a WEP code in order to access the internet. People now use the 128-bit WEP encryption due to the lack of security found in the 64-bit encryption. Think of this WEP code as a password to enter a private server on the internet. Radio waves and access points work together which makes WiFi connection possible. Access points which consist of antennas and routers are the main source which transmit and receive radio waves. Antennas work stronger and have a longer radio transmission with a radius of 300-500 feet which are used in public areas while the weaker yet effective router is more suitable for homes with a radio transmission of 100-150 feet.

Second Semester

In the second semester, it is plan to continue with a detail research of the factors identified before, and also starts developing the application. The implementation of the experiment would also be done based on the blueprint that has been prepared the previous semester. After that, an analysis of the results will be done and from that the author will start formulating the conclusions of the study.

Building WRAPI

During the research phase of the project the author discovered WRAPI. WRAPI is an API developed in C++ for the Windows XP operating system. It provides a set of function calls that can be used to send and receive data to network adaptors. The initial appeal of WRAPI was that it claimed to be "a hardware- independent tool that works with any IEEE 802.11b wireless network hardware vendor" [WRAPIweb]. The source code for this API was also freely available. The WRAPI website provides significant documentation on the features of WRAPI. However, there is less information available for building and using WRAPI. After my experiences with WRAPI the author have created his own guide and included it as an appendix to this report. In this section the author will point out some of the major issues that the author faced in using WRAPI as well as an overview of how WRAPI works.

WRAPI provides WLAN monitoring tools for a Windows XP based system. The WRAPI documentation describes the tools and their implementation as such [WRAPIweb]:

> *The WRAPI software library (wrapi.dll) allows applications running in user space on mobile end stations to query information about the IEEE 802.11 network they are attached to. WRAPI works with any IEEE 802.11b wireless network hardware vendor.*
>
> *WRAPI functions obtain information about the wireless LAN using the NDIS User Mode I/O Protocol (NDISUIO).*

NDISUIO is a connection-less, NDIS 5.1 compliant protocol driver. It allows user-mode applications to establish and tear-down bindings to network adapters (Ethernet, WLAN etc.) Further, it also supports setting packet filters, sending and receiving data, and handling plug-and-play events. Therefore, as an NDIS_aware component, NDISUIO can directly open an NDIS miniport driver (i.e. network card driver) to send requests, set, and query information. NDISUIO provides an interface between a user-mode application and NDIS using DeviceIoControl (similar to the UNIX ioctl). The NDISUIO driver (ndisuio.sys) is already installed in your system under:

C:\WINDOWS\system32\drivers 55

Started with creating a test application, in C++, using the sample code provided by WRAPI (this application and development project are provided as the "app" project in the source code). The first thing the author found out was that the author could not simply include the WRAPI binaries provided on the website because they were built with a different version of the C Runtime Library than the version that was available on his development environment. Any application that will use the WRAPI DLL has to be linked with the same version of the libraries as the DLL was linked against. The cleaner option was to rebuild the DLL on my system and link it with the available library version as opposed to attempting to replace the version of the library on my system. The WRAPI DLL needs to be built by including the "nuiouser.h" file from the Windows XP DDK (Driver Development Kit). It is important that the latest version of the DDK, Windows XP DDK SP1 is used. Previous versions of the DDK will not have the right definitions in the header file that needs to be included; older versions will result in improper runtime behavior.

The older version of the "nuiouser.h" file contained the following define statement:

*#define IOCTL_NDISUIO_OPEN_DEVICE *
_NDISUIO_CTL_CODE(0x200, METHOD_BUFFERED, FILE_ANY_ACCESS)

The current version of the "nuiouser.h" file contains the following statement. This is the version that should be used for compiling WRAPI.

*#define IOCTL_NDISUIO_OPEN_DEVICE *
_NDISUIO_CTL_CODE(0x200, METHOD_BUFFERED, FILE_READ_ACCESS | FILE_WRITE_ACCESS)

The test application first calls WRAPI methods to bind with the correct device and attach to the appropriate network. It then calls a WRAPI method which retrieves the list of visible APs which contains each AP's MAC address and signal strength. All memory to store the list of network devices and the list of APs is allocated and freed inside of the application and not the DLL. Once able to rebuild the DLL then I am able to build the test application by linking it against the new DLL. There are several Visual Studio project options that have to be set in order to build the DLL and the app that calls 56 the DLL. Before running the application the user needs to verify that the NDISUIO windows service has been started and the Wireless Zero Configuration Service has been stopped. *"Wireless Zero Configuration is a Windows service that is normally started at boot time. It provides auto configuration for the 802.11 wireless adapters (NICs) by scanning for available access points and associating with the strongest signal. This service automatically binds to NDISUIO and does not allow other apps/dlls like WRAPI to bind to it*" [WRAPIweb].

Approximation of RSSI measurements

Because of no line of sight restrictions and the effects (reflection and attenuations) due to the person who holds a laptop or PDA device, the equation for signal propagation in free-fields is not suited for the indoor area. Therefore we decide to approximate the correlation between the signal strength and the distance based on measurements in order to improve our estimation.

Therefore, the author uses the value of the RSSI Capture using WRAPI to get a correlation to the distance between sender and receiver in the network [1]. The RSSI value is providing the distance between the received signal strength and an optimal receiver power rank so called *the golden receiver power rank.* The definition of the golden power receiver rang is displayed by Figure 2.1.

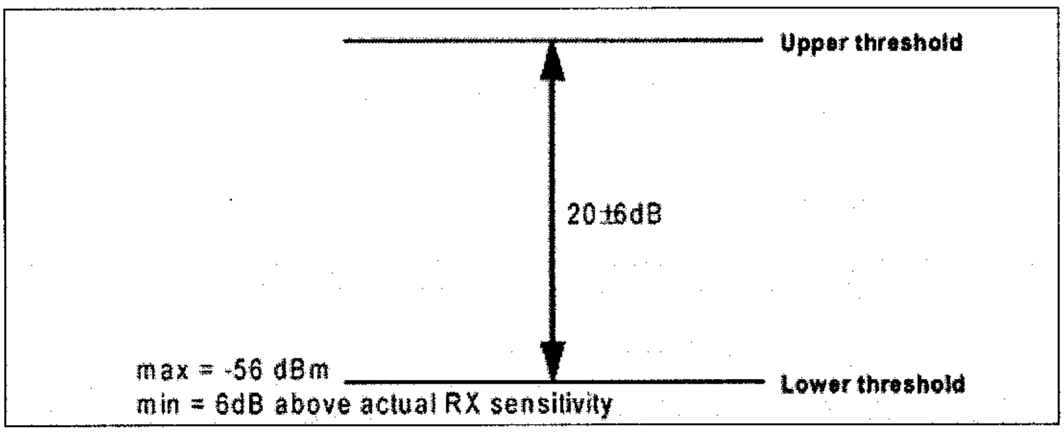

Figure 2.1: Golden receiver power rank of the RSSI

The golden receiver power rank is limited by two thresholds. The lower threshold is clearly defined by an offset of 6dB to the actual sensitivity of the receiver. The maximum of this value is predefined by -56dBm. The upper threshold is 20dB over the lower one. The accuracy of the upper threshold is about ±6dB.

Assume S assign the received signal strength, the value of S is determined by:

$S = RSSI + T_0$, for $RSSI > 0$
$S = RSSI - T_u$, for $RSSI < 0$
$T_0 = T_u + 20dB$

Where T_0: upper threshold,
T_u: lower threshold.

The definition of the golden receiver rank limits the conversion of the RSSI to a distance. If the value of the RSSI is in the golden receiver range defined by zero no unique function can be approximated. Therefore only measurements which results in a positive range of the RSSI can be considered by a functional approximation.

Process Flow

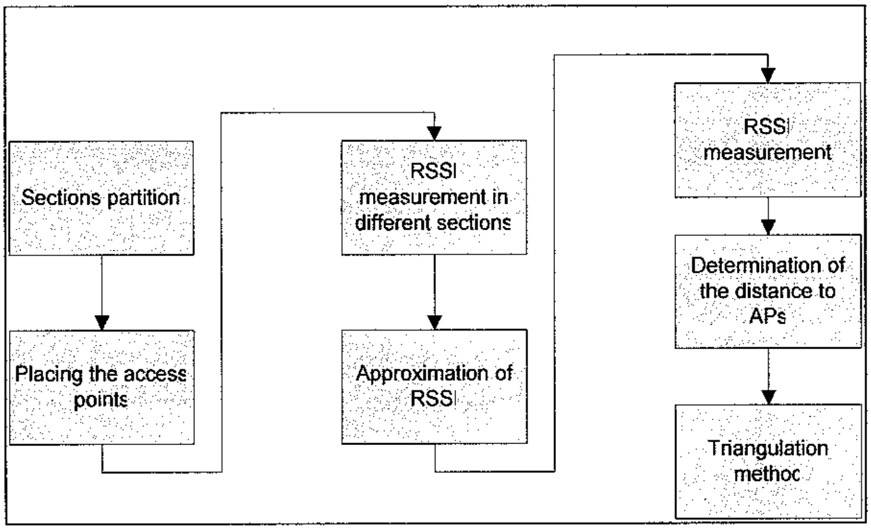

Figure 3.1: Flow chart of wifi-based positioning system

The prototype implementation of the Wifi based positioning system can be divided into several phases. Firstly, the considered indoor area is section partition and then the access point are placed; next, the RSSI function which is fitting the environmental conditions is approximated. Then the location method is used in order to determine the position estimation of a mobile terminal based on RSSI-measurements. In this context it involves the so called triangulation method by applying the mathematical calculations on the RSSI value of three access point. The flow chart of the system illustrated by Figure 3.1 presents an overview of the operation breakdown.

3.2 TOOLS / SOFTWARES

The tools and software that will be used to complete this project would be:

Resources – Internet, Text Books, Journals

These resources will be used to gather detailed information and find the previous works which are related to the topic. Main access would be from the Information Resource Center as well as the Residential College.

Wireless Equipment – Access point, Router, wireless card, wireless base station

These equipments used to create network that broadcast signals to wireless equipped devices (router) and accept the wireless signal and relay information (Wifi card).

Access point

Is a device that transports data between a wireless network and a wired network (infrastructure). For this project, 3 access points are being used. All the access point are WRT54G model manufactured by Linksys.

Router

A router is a freestanding piece of hardware that transfers data between networks or between networked PCs and the internet.

Wireless card

A wireless card is an internal or external card for your notebook that allow it to connect wirelessly to a wireless-enable network in the office, coffee shop, bookstore air port or your home. For this project I am using the wireless G notebook adapter (DWL-G630) manufactured by D-Link.

Wireless base station

Is an Intel PRO/Wireless 2200 network connection 802.11b/g access point with built-in internet router. With a connection to a cable or DSL modem, the base station can give both wired and wireless computers simultaneous access to the internet.

Visual Studio C++

Programming software used to build and compile program. The programming language used is C++ language

CHAPTER 4

RESULTS AND DISCUSSION

The test program developed using the sample provided able to detect the access points available. The parameter that can be get using the program including the SSID, MAC address, Network type, WEP, RSSI and etc. The source codes of the test program are included at the appendices section.

Figure 4.1 to figure 4.4 which also at the appendices section shows the output of the program from starts until it shows the result.

The developed system is based on the known and already published triangulation methods using the received signal power strength of the surrounding access points. For precise position estimation, the dependence between the distance and the received signal strength has to be determined. Especially in the indoor area, boundary conditions like reflections and wall damping make the use of the equation for the free-field propagation impossible. Therefore, the required calculations of the distances are estimated by an approximation of the *Received signal Strength Indicator* (RSSI)

Measurement

The implemented system has been tried out in an indoor scenario consisting of three wifi access point and a laptop which provides the location system. The wifi access points are realized by notebooks equipped with the wireless G notebook adapter card (DWL-G630) manufactured by D-Link. The access points have been distributed in a room with a size of $48m^2$. The room is portioned in 1m x 1m sections or squares. The scenario is illustrated by Figure 5.1.

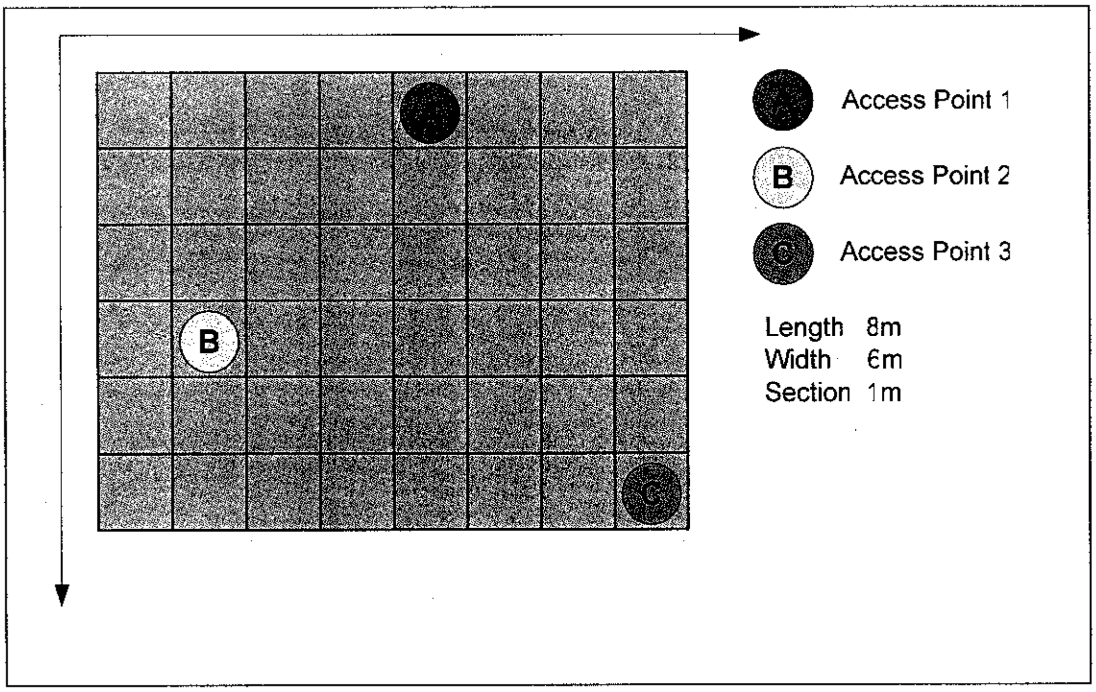

Figure 5.1: Example of Wifi scenario

To obtain a good approximation function between the RSSI and the single access point distances in a number of randomly chosen sections several measurements have been recorded. The mean of all measurements belonging to one section and access point has formed the reference values of the RSSI in a section.

The next step is to determine the distance of the subject from the access points (AP's). For that purpose, the RSSI of the AP's will be used.

To know the distance testing have been made based on the RSSI gathered. The testing is made by calculating manually the distance of the location based on the RSSI collected. The testing is done in the building and the data collected shows as below: (See Table 2.1 and Figure 6.1)

Table 2.1: RSSI and distance (m)

RSSI (db)	DISTANCE (m)
-32	10
-35	8.5
-40	6.5
-45	5.5
-47	5
-56	2
-58	1.8
-65	0.8
-70	0

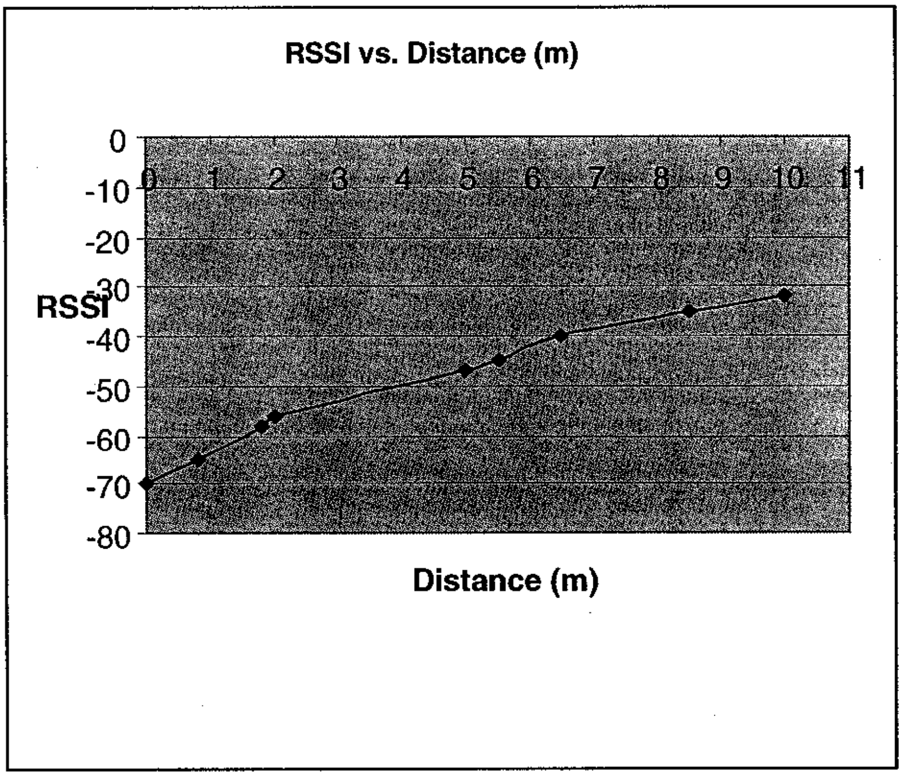

Figure 6.1: Graph RSSI vs. Distance (m)

When the distances are identified, it will be the value of the radius of the AP available. Then, the triangulation can be made and the intersections of all the circle is the location that want to be discover. (See Figure 7.1) While for Figure 8.1 to Figure 8.5 located at appendices are the interface and the output of the completed system.

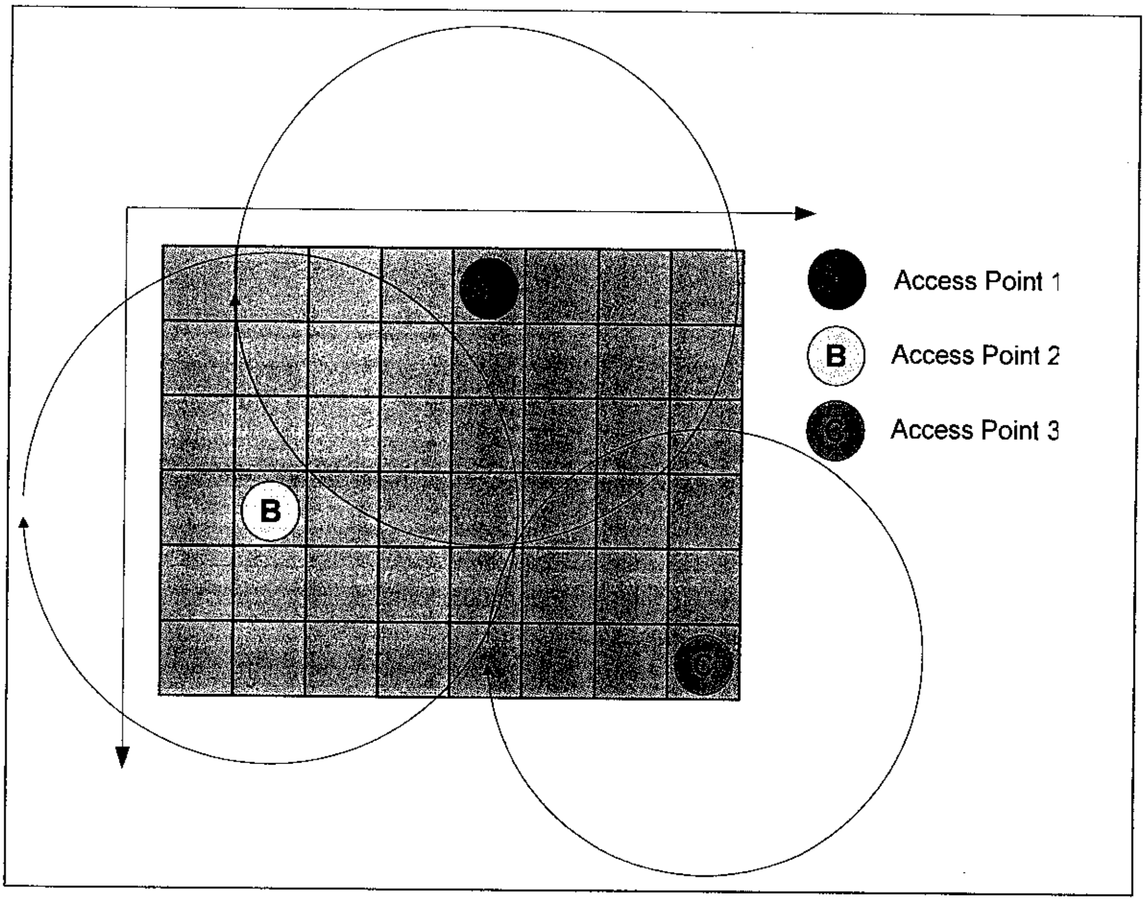

Figure 7.1: Intersection of three circles that make triangulation

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

Although there are several locations detection systems developed with various methods such as using Ultra sound, Radio Frequency (RF), Infra Red (IR) and so on but still there are flaws that can be improved. Moreover the existing system also not specifically designed for the working conditions of emergency network and thus generally unsuitable for this purpose. It is hoped that with some improvement to the proposed system able to work in the emergency condition.

In this paper, I have presented an indoor positioning system based on signal strength measurements, which were approximated by the received RSSI in a mobile device. The triangulation method combined with least square estimation is used to predict the position of the terminal. The functional dependence between the received RSSI and the distance was got by a well fitted polynomial approximation. This function is general restricted by a positive sign of the RSSI. It is expected that the accuracy of the estimation could be increased by combining the method with the results of other kinds of location estimation.

5.2 RECOMMENDATION

Further work can be done in order to improve the accuracy of locations detected. Other research can be done for the approach in determining the location for multi level building. Moreover a system can be setup based on client server for location detection.

REFERENCES

Adams, N., Gold, R., Schilit, Bill N., Tso, Michael M., Want, R. (1993), "The *ParcTab Mobile Computing System*", Palo Alto Research Center, Xerox Corporation, California, USA.

Bahl P., Padmanabhan V. (2000), *"RADAR: An In-Building RF-based User Location and Tracking System"*, Microsoft Corporation, Washington, USA.

Bekris, Kostas E., Kavaki, Lydia E., Ladd, Andrew M., Marceau, G., Rudys, A., Wallach, Dan S. (2002), *"Robotics-Based Location Sensing using Wireless Ethernet"*, Department of Computer Science, Rice University, Texas, USA.

Bhasker, E, Boyer, R., Brown, Steven W., Griswold, William G., Jay, Gregory R., Shapiro, R. Benjamin (2002), *"Active Campus- Sustaining Educational Communities through Mobile Technology"*, Department of Computer Science and Engineering, California, USA.

Borriello, G., Hightower J., Vakili, C., Want R. (2001), *"Design and Calibration of the SpotON Ad-hoc Location Sensing System"*, Intel Research, California, USA.

Chakraborty, A., Balakrishnan, H., Priyantha, Nissanka B. (2000), "The Cricket Location-Support System", In: *"6th ACM International Conference on Mobile Computing and Networking", August 2000, Boston,* Massachusetts, USA.

Falcao, V., Gibbons, J., Hopper, A., Want, R. (1992), *"The Active Badge Location System*", Olivetti Research Ltd., Cambridge, England.

Hotspot,(2003) (www.webopedia.com), Available:
http://www.webopedia.com/TERM/h/hotspot.html 2003

Mark, Jon W. Wireless Communications and Networking. Upper Saddle River, NJ: Prentice Hall, 2003.

White, Ron. How Computers Work. 7th ed. Ed. Charlotte Clapp. Indianapolis, IN: QUE, 2003.

Dornan, Andy. The Essential Guide to Wireless Communications Applications: From Cellular Systems to Wi-Fi. 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2002.

Brain, Marshall. "How WiFi Works." How Stuff Works. 3 Dec. 2004 <http://computer.howstuffworks.com/wireless-network.htm>

"How WiFi Works?" CNN. 5 Dec. 2004 http://www.cnn.com/interactive/tech/0303/wifi.explainer/content1.html

APPENDICES

Test Program coding to detect access point

```
#include <iostream>
#define INCLUDE_WINDOWS_H
#include <curses.h>
#include <mmsystem.h>
#include "stdafx.h"
#include "WrapiExports.h"

#include <process.h>


unsigned __stdcall realmain (void * param){


initscr();
keypad(stdscr, TRUE);
echo();
nonl();
cbreak();
int row, col;
getmaxyx(stdscr,row,col);


        WRAPI_NDIS_DEVICE                   *pDeviceList = NULL;
        HRESULT                             hRes;

        long                    lItems = 0;
        long                    lNumItems = 0;
        UCHAR                               Ssid[32] = {0};

        AP_DATA                             *pAP_list = NULL;
        CHAR                    *pSSId = "UCSD";

        ULONG                               lSSIdLength = strlen(pSSId);
        ULONG                               lRTSThresh = 0;

        MAC_ADDR                addr = {0};
        DOT_11_STATS                        DOT11Stats;

        /* Get the list of Devices in the System */
        WRAPIInitInstance();
        hRes = WRAPIEnumerateDevices(&pDeviceList, &lItems);
printf("%ul (should be 01)\n",hRes);
system("pause");
clear();
        /* Print the list of devices obtained */
```

```
	for ( int i = 0; i < lItems; i++)
	{

		printw("\t%d) %ws\n", i+1, pDeviceList[i].pDeviceDescription);
	}

	printw("\n\tPlease select your WiFi card and press enter: ");
	int select;
	scanw("%d", &select);

	/* Open the device at pDeviceList[1] -- pDeviceList[0] happens to be the

 * Ethernet Interface */
	hRes = WRAPIOpenNdisDevice(pDeviceList[select-1].pDeviceName);

	printf("%x\n",hRes);
	/* Get the SSID Of the Network */
	hRes = WRAPIGetSSId(Ssid);

	/* Set the network SSID to "UCSD" */
	//hRes = WRAPISetSSId((UCHAR*)pSSId, lSSIdLength);

	/* Get the value of RTS Threshold */
	hRes = WRAPIGetRTSThreshold(&lRTSThresh);

	/* Get the MAC address of the AP to which this station is associated */
	hRes = WRAPIGetAssociatedAP(addr);

	/* Get Packet-level statistics from the 802.11 interface */
	hRes = WRAPIGetPacketStats(&DOT11Stats);

	/* Get a list of all APs within range of this station in a list of AP_DATA
	 * structures */
float r=0;
	while (1){

		r+=1.0;
		//DWORD  time = timeGetTime  ();

	hRes = WRAPIGetAPList(&pAP_list, &lNumItems);
	//	DWORD  time2 = timeGetTime();
//mvprintw(20,50,"\ttime:\t%ld\n", time2-time);
	/* Print out the list of all APs and their signal strengths */

	for ( i = 0; i < lNumItems && i < 3; i++ )
	{


		mvprintw(i*7+0,0,"\tAP #%d:\t%s\n", i+1, pAP_list[i].Ssid);
		mvprintw(i*7+1,0,"\tMAC Address:\t");
		for ( int j = 0; j < 5; j++)
```

```
{
printw("%02x:", pAP_list[i].mac_addr[j]);
}
printw("%02x\n", pAP_list[i].mac_addr[j]);

UINT maxrate=0;
for ( j = 0; j < 16; j++)
{
        INT rate = pAP_list[i].SupportedRates[j];
        rate &= 0x7f;
if(rate > maxrate)

        maxrate = rate;
}

mvprintw(i*7+2,0,"\tMax Rate:\t%.1f Mbps\n", maxrate*0.5f);

mvprintw(i*7+3,0,"\tRSSI:\t\t%d dBm\n", 100 + pAP_list[i].Rssi);

std::string WEP[2];
WEP[0] = "None";
WEP[1] = "Required";

mvprintw(i*7+4,0,"\tWEP:\t\t%s\n", WEP[pAP_list[i].Privacy].c_str());

std::string NetworkType[4];
NetworkType[0] = "FH";
NetworkType[1] = "DS";
NetworkType[2] = "5G";
NetworkType[3] = "2.4G";

mvprintw(i*7+5,0,"\tNetworkType:\t%s\n",

NetworkType[pAP_list[i].NetworkTypeInUse].c_str());
mvprintw(i*7+6,0,"\tFreq:\t\t%.2f MHz\n", pAP_list[i].Freq);


std::string Mode[3];
Mode[0] = "AdHoc";
Mode[1] = "InfraStructure";
Mode[2] = "AutoUnknown";

mvprintw(i*7+7,0,"\tMode:\t\t%s\n",

Mode[pAP_list[i].InfrastructureMode].c_str());

}
//beep();
refresh();
Sleep(100);
clear();
```

```
        free(pAP_list);
        }

        for ( i = 0; i < lItems; i++ )
        {
                free(pDeviceList[i].pDeviceName);
                free(pDeviceList[i].pDeviceDescription);
        }

        free(pDeviceList);
        return 0;

}

int main(int argc, char* argv[])
{
realmain(NULL);

HANDLE hThreads[1];
unsigned *threadH = NULL;
hThreads[0] = (HANDLE)_beginthreadex(NULL, 0,realmain,NULL, 0, threadH);
if(hThreads[0]){

        std::cout << hThreads[0] << std::endl;
WaitForMultipleObjects(1,
                hThreads,
                TRUE,
                INFINITE);
}else{
printf("error\n");
}

        return 0;
}
```

Output

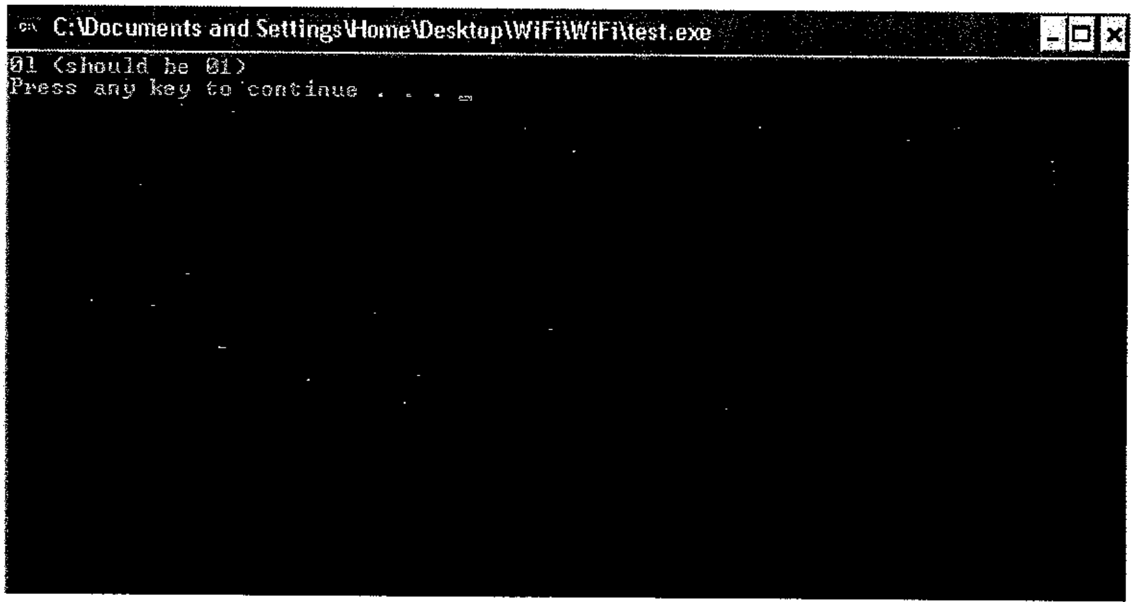

Figure 4.1: Test Program - First screen

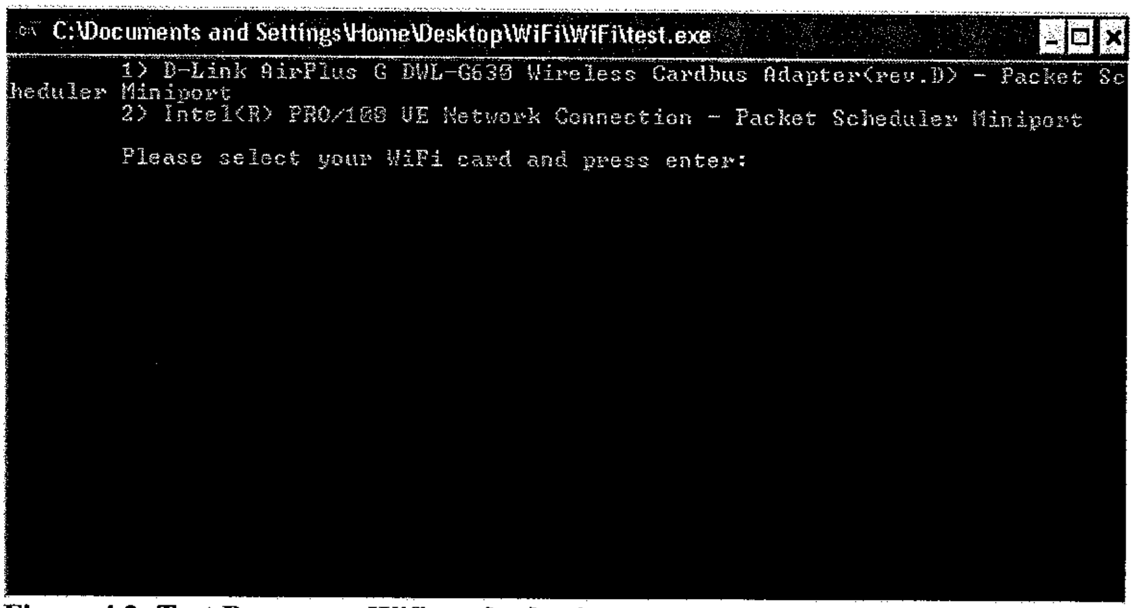

Figure 4.2: Test Program - Wifi card selection screen

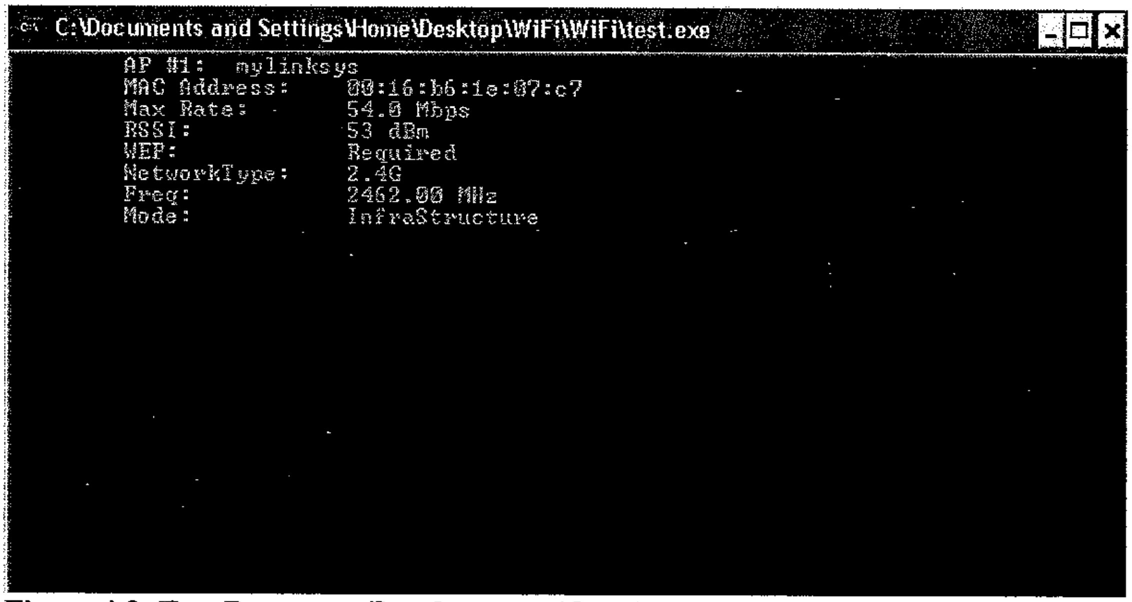

Figure 4.3: Test Program - Result screen (One Access Point)

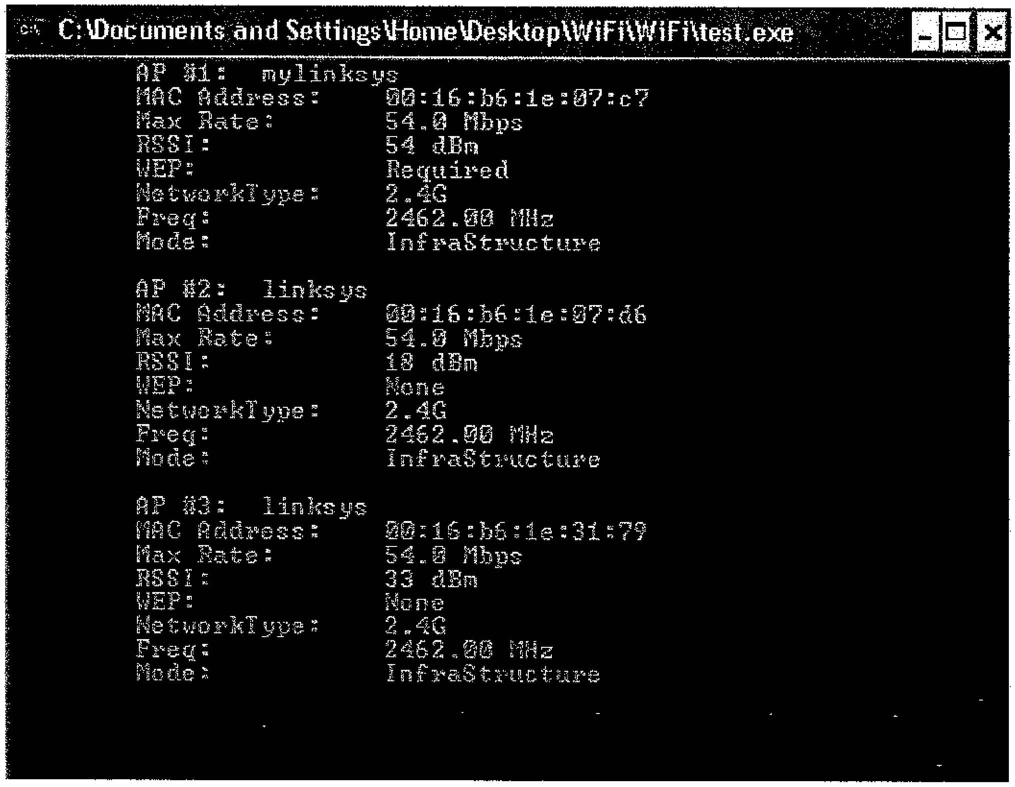

Figure 4.4: Test Program - Result screen (Three Access Point)

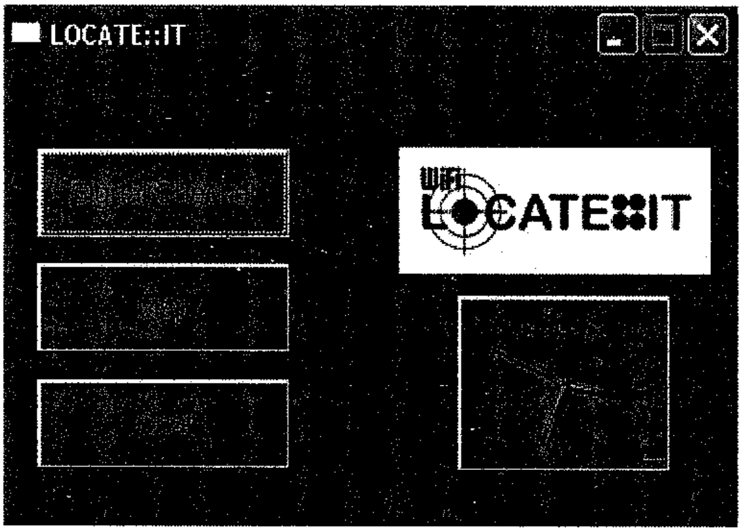

Figure 8.1: Main Page

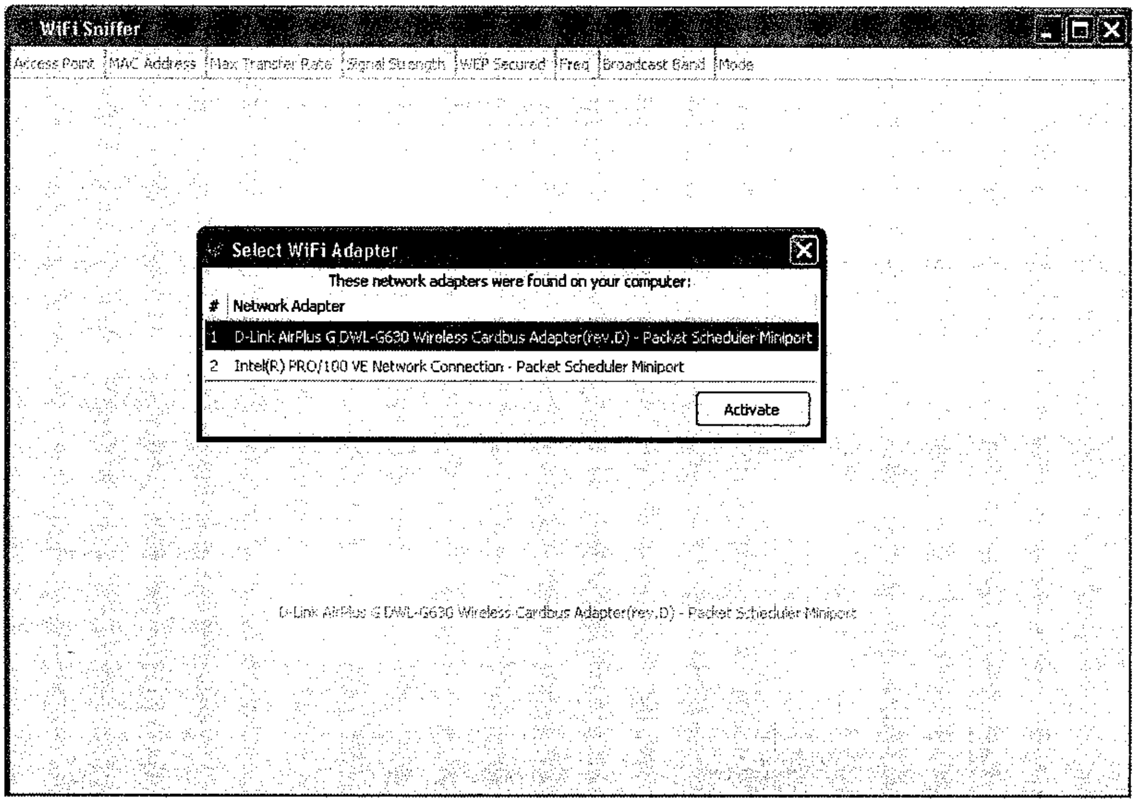

Figure 8.2: Wifi Adapter Selection

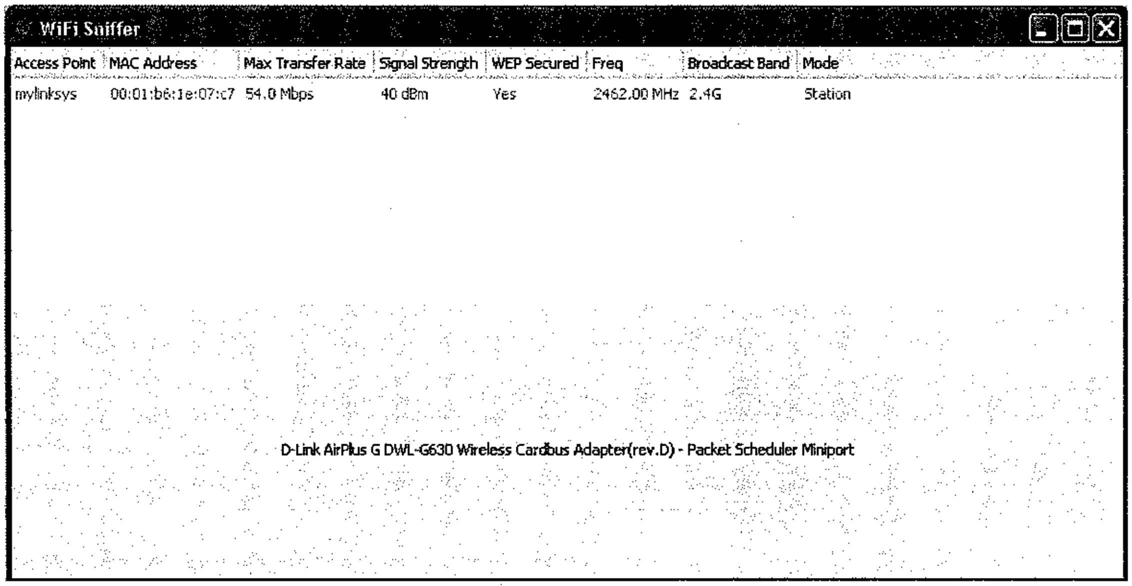

Figure 8.3: Result of available access point

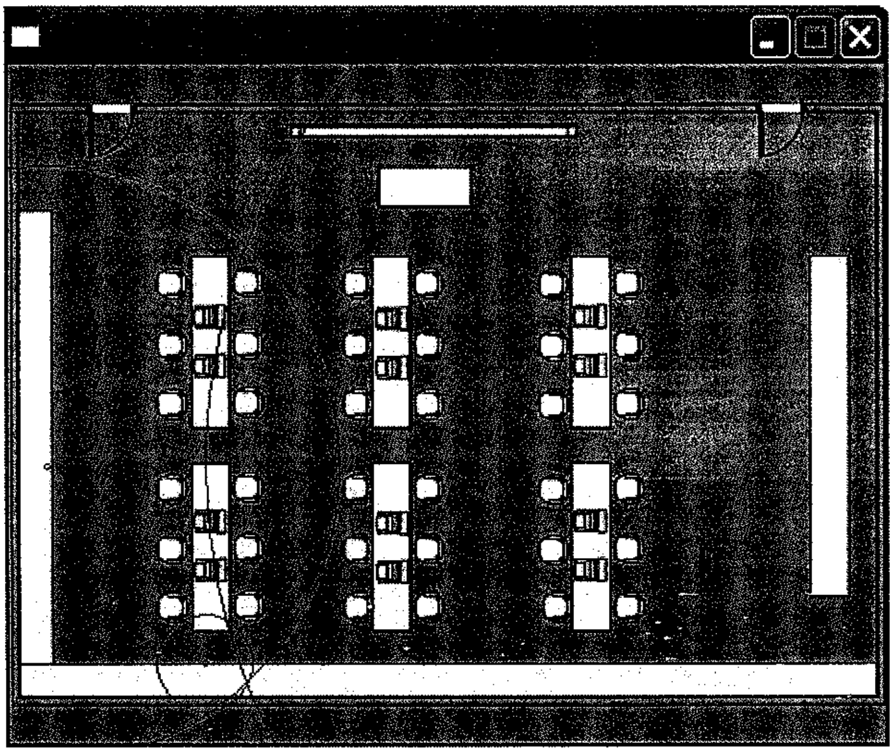

Figure 8.4: Map and triangulation

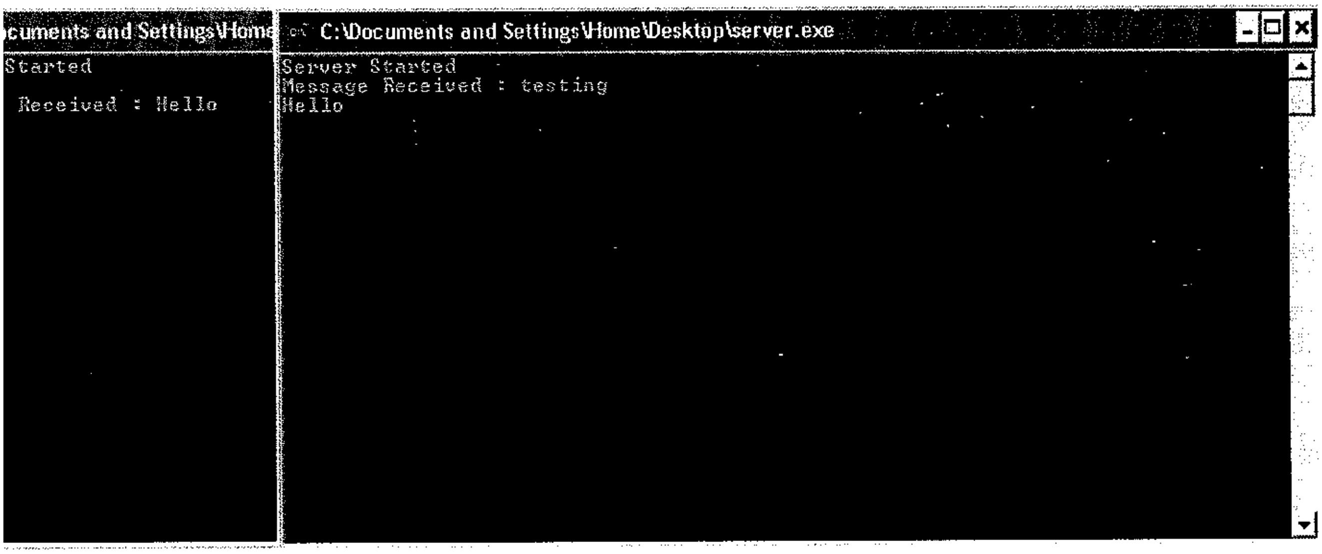

Figure 8.5: Messaging between client and server

Main Page

```
#include "button.h"
#include <stdio.h>
#include <stdlib.h>


void Button_Chk(Fl_Widget*, void*) {
 printf("Check Signal!!\n");
        system ("wifi.exe");
}

void Button_Load(Fl_Widget*, void*) {
 printf("Load Map\n");
 system ("loadmap.exe");
}

void Button_Chat(Fl_Widget*, void*) {
 printf("Chat\n");
 system ("client.exe");

}

#include <FL/Fl_Image.H>
static unsigned char idata_locateitsmall[] =
{255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,254,255,
255,254,255,255,254,255,255,252,255,255,252,255,255,251,255,255,254,255,255,254,
255,255,255,254,255,255,254,255,255,255,255,255,255,255,254,255,253,254,255,253,
252,255,253,252,255,253,255,255,253,255,255,253,255,254,253,255,255,253,255,255};
static Fl_RGB_Image image_locateitsmall(idata_locateitsmall, 160, 63, 3, 0);

int main(int argc, char **argv) {
 Fl_Double_Window* w;
 { Fl_Double_Window* o = new Fl_Double_Window(371, 231, "LOCATE::IT");
  w = o;
  o->color((Fl_Color)179);
  { Fl_Button* o = new Fl_Button(15, 44, 130, 45, "Signal Checker");
   o->color((Fl_Color)181);
   o->selection_color((Fl_Color)199);
   o->callback((Fl_Callback*)Button_Chk);
  }
  { Fl_Button* o = new Fl_Button(15, 101, 130, 45, "Map");
   o->color((Fl_Color)181);
   o->selection_color((Fl_Color)199);
   o->callback((Fl_Callback*)Button_Load);
  }
  { Fl_Button* o = new Fl_Button(15, 159, 130, 45, "Chat");
   o->color((Fl_Color)181);
   o->selection_color((Fl_Color)199);
   o->callback((Fl_Callback*)Button_Chat);
```

```
  }
  { Fl_Clock* o = new Fl_Clock(230, 117, 110, 88);
   o->color((Fl_Color)181);
  }
  { Fl_Box* o = new Fl_Box(200, 45, 160, 58);
   o->image(image_locateitsmall);
  }
  o->end();
 }
 w->show(argc, argv);
 return Fl::run();
}
```

Adapter Dialog

```
#include "test.h"

 void
OKbutton_adapterList_clicked_cb (GtkButton *button, gpointer data)
{
        printf("OK button clicked\n");
        GtkTreeIter iter;
        GtkTreeSelection *selectHandler =
gtk_tree_view_get_selection(GTK_TREE_VIEW(adapterDialog.treeView_adapterList));
   GtkTreeModel
*model=gtk_tree_view_get_model(GTK_TREE_VIEW(adapterDialog.treeView_adapterList));
        gchar *adapter;

                guint index;
        if (gtk_tree_selection_get_selected (selectHandler, &model, &iter))
     {
          gtk_tree_model_get (model, &iter,0,&index, 1, &adapter, -1);


//          adapte = g_strdup_printf("%s%s",adapte,adapter);

                                //gtk_tree_model_get (model, &iter, 0, &adapter, -1);

        gtk_label_set_text(GTK_LABEL(mainWindow.label_status),adapter);
          g_free (adapter);

                                hRes = WRAPIOpenNdisDevice(pDeviceList[index-
1].pDeviceName);

                //error handling here
                        printf("Adapter opened successfully\n");
                //open_Adapter(pDeviceList[index-1].pDeviceName);
```

```
    }

gtk_widget_destroy(adapterDialog.dialog_main);
gtk_widget_set_sensitive (mainWindow.window_main,TRUE);
printf("windows resize enable\n");
//gtk_window_set_resizable       (GTK_WINDOW(mainWindow.window_main),TRUE);


printf("created APthread\n");
//update_AP(NULL);
 APupdatethread = g_thread_create(thread_update_AP,NULL,TRUE,NULL);

 APGUIupdatethread = g_thread_create(thread_update_APGUI,NULL,TRUE,NULL);

 APScanupdatethread = g_thread_create(thread_update_APscan,NULL,TRUE,NULL);




//gtk_timeout_add (300,thread_update_AP,NULL);
//gtk_timeout_add (300,thread_update_APGUI,NULL);
/*

hRes = WRAPIOpenNdisDevice(pDeviceList[index-1].pDeviceName);

        hRes = WRAPIGetSSId(Ssid);

        //hRes = WRAPISetSSId((UCHAR*)pSSId, lSSIdLength);

        hRes = WRAPIGetRTSThreshold(&lRTSThresh);
        hRes = WRAPIGetAssociatedAP(addr);

        hRes = WRAPIGetPacketStats(&DOT11Stats);

enum
{
  TITLE_COLUMN,
  N_COLUMNS
};


 GtkListStore *store;
 // GtkWidget *tree;
  GtkTreeViewColumn *column;
  GtkCellRenderer *renderer;

  store = gtk_list_store_new (1, G_TYPE_STRING );
int i;
```

```
 GtkTreeIter iter1;


for (  i = 0; i < 2; i++)
        {
char tmp[100];
sprintf(tmp,"haha");
gtk_list_store_append (store, &iter1);
gtk_list_store_set (store, &iter1, TITLE_COLUMN , tmp,  -1);

        }

  mainWindow.treeView_APlist = gtk_tree_view_new_with_model (GTK_TREE_MODEL
(store));

 g_object_unref (G_OBJECT (store));
  renderer = gtk_cell_renderer_text_new ();
  g_object_set (G_OBJECT (renderer),
          "foreground", "red",
          NULL);

        renderer = gtk_cell_renderer_text_new ();
column = gtk_tree_view_column_new_with_attributes ("AP List", renderer,
                                   "text", 0,
                                   NULL);

  gtk_tree_view_append_column (GTK_TREE_VIEW (mainWindow.treeView_APlist),
column);




GtkWidget *dud = gtk_button_new();
;

        gtk_box_pack_start (GTK_BOX (mainWindow.frame_APlist),
              mainWindow.treeView_APlist, TRUE, TRUE, 0);


   gtk_widget_show (mainWindow.treeView_APlist);
        // gtk_widget_show (tree);


select = gtk_tree_view_get_selection (GTK_TREE_VIEW (mainWindow.treeView_APlist));

gtk_tree_selection_set_mode (select, GTK_SELECTION_SINGLE);

g_signal_connect (G_OBJECT (select), "changed",
           G_CALLBACK (tree_selection_changed_cb),
           NULL);
```

```
//gtk_box_pack_start (GTK_BOX (APlist),
 //              tree, TRUE, TRUE, 0);

//gtk_box_pack_start (GTK_BOX (APdetails),
  //             tree, TRUE, TRUE, 0);

gtk_container_add (GTK_CONTAINER (mainWindow.frame_main),
mainWindow.frame_APlist);
gtk_container_add (GTK_CONTAINER (mainWindow.frame_main),
mainWindow.frame_APdetails);
gtk_container_add (GTK_CONTAINER (mainWindow.window_main),
mainWindow.frame_main);
  //gtk_widget_show(dud);
       //gtk_widget_show(APlist);
       //gtk_widget_show(APdetails);
       //gtk_window_resize          (GTK_WINDOW(window),
   //                        400,
   //                        600);

//GTimer *my time =g_timer_new( );

GThread*   APupdatethread = g_thread_create(APupdate,NULL,TRUE,NULL);
 */


}



static void
treeView_adapterList_changed_cb (GtkTreeSelection *selection, gpointer data)
{
     GtkTreeIter iter;
     GtkTreeModel *model;
     gchar *adapter;

     if (gtk_tree_selection_get_selected (selection, &model, &iter))
     {
          gtk_tree_model_get (model, &iter, 1, &adapter, -1);


        gtk_label_set_text(GTK_LABEL(mainWindow.label_status),adapter);

          g_free (adapter);
     }


}

void doNothing_adapterDialog_cb(GtkWindow * window, gpointer data){
```

```
printf("doing nothing\n");
}

void init_adapterDialog(){

	//create dialog window
	adapterDialog.dialog_main = gtk_dialog_new();

	//setup dialog window
	gtk_window_set_title  (GTK_WINDOW(adapterDialog.dialog_main),"Select WiFi
Adapter");
	gtk_window_set_modal (GTK_WINDOW(adapterDialog.dialog_main),true);
	gtk_window_set_transient_for
(GTK_WINDOW(adapterDialog.dialog_main),GTK_WINDOW(mainWindow.window_main));


	//gtk_window_set_type_hint();
	//setup label
/*
	GdkWindowAttr adapterDialogAttr;
	adapterDialogAttr.title="WiFi Sniffer";
	adapterDialogAttr.x =5;
	adapterDialogAttr.y=2;
 GdkWindowType window_type;
 GdkCursor *cursor;
 gchar *wmclass_name;
 gchar *wmclass_class;
 gboolean override_redirect;
*/

	adapterDialog.label = gtk_label_new("These network adapters were found on your
computer:");

	//setup TreeView for Adapter selection

		//setup list
	GtkListStore *store = gtk_list_store_new (2,   G_TYPE_UINT, G_TYPE_STRING );
	adapterDialog.treeView_adapterList = gtk_tree_view_new_with_model
(GTK_TREE_MODEL (store));
	g_object_unref (G_OBJECT (store));

		//setup cell renderer
	GtkCellRenderer *renderer = gtk_cell_renderer_text_new ();
	g_object_set (G_OBJECT (renderer),"foreground", "red",NULL);

		//setup column
	GtkTreeViewColumn *column;

		//setup selection handler
	GtkTreeSelection *selectHandler = gtk_tree_view_get_selection (GTK_TREE_VIEW
(adapterDialog.treeView_adapterList));
```

```
	gtk_tree_selection_set_mode (selectHandler, GTK_SELECTION_SINGLE);
	g_signal_connect (G_OBJECT (selectHandler), "changed",G_CALLBACK
(treeView_adapterList_changed_cb),NULL);


	column = gtk_tree_view_column_new_with_attributes ("#", renderer,"text", 0,NULL);
	gtk_tree_view_append_column (GTK_TREE_VIEW
(adapterDialog.treeView_adapterList), column);

	column = gtk_tree_view_column_new_with_attributes ("Network Adapter",
renderer,"text", 1,NULL);
	gtk_tree_view_append_column (GTK_TREE_VIEW
(adapterDialog.treeView_adapterList), column);

	//setup OK button
	adapterDialog.responsebutton = gtk_button_new ();
	gtk_button_set_label(GTK_BUTTON(adapterDialog.responsebutton),"Activate");
  //printf("responsebutton actionHandler: %d\n",adapterDialog.responsebuttonAction);


	//add label,TreeView,button to Dialog window
	gtk_box_pack_start (GTK_BOX (GTK_DIALOG (adapterDialog.dialog_main)-
>vbox),adapterDialog.label, TRUE, TRUE, 0);
	gtk_box_pack_start (GTK_BOX (GTK_DIALOG (adapterDialog.dialog_main)-
>vbox),adapterDialog.treeView_adapterList, TRUE, TRUE, 0);
	gtk_box_pack_start (GTK_BOX (GTK_DIALOG (adapterDialog.dialog_main)-
>action_area),adapterDialog.responsebutton, TRUE, TRUE, 0);
	gtk_widget_show (adapterDialog.label);
	gtk_widget_show (adapterDialog.treeView_adapterList);
	gtk_widget_show (adapterDialog.responsebutton);


 /*
int i;
 GtkTreeIter iter1;
for ( i = 0; i < lItems; i++)
	{
char tmp[100];
sprintf(tmp,"%ws",pDeviceList[i].pDeviceDescription);
gtk_list_store_append (store, &iter1);
gtk_list_store_set (store, &iter1,INDEX, i+1, TITLE_COLUMN , tmp,  -1);
	}
*/


}

void show_adapterDialog(){

	Hss = fopen("salam.txt","a");
	fprintf(Hss,"showing adapter box\n");
```

```
fclose(Hss);

        gboolean reallyshowDialog;
         GtkTreeIter iter;
GtkTreeModel
*model=gtk_tree_view_get_model(GTK_TREE_VIEW(adapterDialog.treeView_adapterList));


        //gtk_tree_model_get_iter_first(model,&iter);

        if(!gtk_tree_model_get_iter_first(model,&iter)){

        reallyshowDialog=0;
Hss = fopen("salam.txt","a");
        fprintf(Hss,"badh \n");
fclose(Hss);

        }else{

                Hss = fopen("salam.txt","a");
        fprintf(Hss,"good \n");
fclose(Hss);
reallyshowDialog=1;
                while(gtk_tree_model_iter_next (model, &iter));
                printf("%d\n",iter.user_data);
                //if(iter.user_data){}

        }


        if(!reallyshowDialog){

                gtk_label_set_text(GTK_LABEL(adapterDialog.label),"Error:No network
adapters detected.");
gtk_button_set_label(GTK_BUTTON(adapterDialog.responsebutton),"Exit");
gtk_widget_hide (adapterDialog.treeView_adapterList);

//g_signal_handler_disconnect(G_OBJECT
(adapterDialog.responsebutton),adapterDialog.responsebuttonAction);
adapterDialog.responsebuttonAction =g_signal_connect (G_OBJECT
(adapterDialog.responsebutton), "clicked",G_CALLBACK (destroy_mainWindow_cb),NULL);

        }else{
g_signal_connect (G_OBJECT (adapterDialog.dialog_main), "delete-event", G_CALLBACK
(doNothing_adapterDialog_cb), NULL);
adapterDialog.responsebuttonAction = g_signal_connect (G_OBJECT
(adapterDialog.responsebutton), "clicked",G_CALLBACK
(OKbutton_adapterList_clicked_cb),NULL);
        }
gtk_widget_set_sensitive (mainWindow.window_main,FALSE);
gtk_widget_show (adapterDialog.dialog_main); }
```